Spring 4-1-2005

A Hot Rod Shop on the Internet Super Highway

Bill DeKnikker
Dakota State University

A Hot Rod Shop on the Internet Super Highway

Bill DeKnikker

A project submitted in partial fulfillment of the requirements for the
Master of Science in Information Systems
Dakota State University
Spring 2005

DAKOTA STATE dsu UNIVERSITY

MSIS

PROJECT APPROVAL FORM

Student Name: Bill DeKnikker

Expected Graduation Date: May 7 2005

Master's Project Title: A Hot Rod Shop on the Internet Super Highway

Date Project Plan Approved: Fall 2004

Date Project Coordinator Notified and Grade Submitted: April 29 2005

Approvals/Signatures:

Student: _____ Date: 4-29-05

Faculty supervisor: _____ Date: 4-29-05

Committee member: _____ Date: 4/29/05

Committee member: _____ Date: 4/29/05

Revised 1/28/03

Abstract

Hwy 34 Customs, a relatively new custom auto body and repair business in the Madison area wanted to establish a web presence where their products and services could be advertised to a wide variety of people and businesses on the Internet.

After an agreement was made, work began on the design and implementation of a database driven web site, which provides a user-friendly interface that anyone with an administrative user name and password can access. From within the admin controls side of the web site, an admin person can add, edit and delete inventory items and information, including pictures to display on the public side of the web site.

The driving force behind the web site is ASP.net and stored procedures written in Transact SQL. There are also a few basic HTML web pages, which were build in Macromedia's DreamWeaver. DreamWeaver is also used as the FTP agent for uploading the files to a virtual domain server. The virtual domain server is a Windows 2003 server and also includes 50MB of Microsoft SQL Server 2000.

A person with an administrative user name and password can log on to the Admin Controls side of the web site where he can access the various areas of the site. Each area of the Admin Controls side of the web site offer forms, which are used to interact with the SQL Server 2000 database. The ASP.net web forms collect information using parameters supplied by filling out text boxes and pass those parameters to stored procedures on the SQL Server 2000 database (Bennet Wm. McEwan, David Solomon 1997). The stored procedures use the parameters to update the database tables. More ASP.net web pages pull the information back out of the database and display the information to users on the public side of the web page.

Table of Contents

List of Tables

List of Figures

I. Introduction

Hwy 34 Customs and Collision is a full service body shop that specializes in Custom and restoration projects. They have capabilities to fabricate body parts, do complete frame off restorations, design and create custom interiors, overhaul and rebuild all mechanical components, or fix your latest fender bender. They provide custom paint jobs of all kinds. If it's being done, they can do it. And best of all, they do it for South Dakota prices.

Hwy 34 Customs is a fairly new business in the Madison area. Dave Theilbar, owner of PSI Inc. of Madison, SD wants a database driven web site that he can use to advertise the products and services of Hwy 34 Customs.

www.hwy34customs.com will give anyone with administrative rights to the web site the ability to add, update or delete records in an SQL database. These records are used by the front end of the web site to display the products and services offered by Hwy 34 Customs including:

- Stuff For Sale
- Cars For Sale
- Customer's Cars
- Upholstery
- Current Projects
- Future Projects

Other areas of the web site include general information displayed in static HTML web pages. These areas include:

- About Us
- Contact Us
- Request a Quote
- Meet The Staff

Through the use of a web-based interface, an administrator can log on to the web site and add, edit or delete pictures and information about the offered products and services. Before the web site was created, the only way to distribute this information, including the pictures was to purchase expensive ads in news papers and magazines. Now through the use of technology this information can be presented to a world-wide audience on-demand. The web site can be updated as often as necessary, 24 hours a day, 7 days a week and the changes are immediate.

The Hwy 34 Customs web site is database driven using a sever side SQL Server 2000 database set up on a virtual domain host. The web site itself was developed using Microsoft Visual Studio.net, which was purchased by the owner of the site for the purpose of development. Along with Visual Studio.net, the owner also purchased SQL Server Development Tools, which includes SQL Query Analyzer and Enterprise Manager. From a user's perspective, the only tools needed to administer the web site is a computer with an Internet connection and a web browser.

II. The Servers and Software

Domain Name

A domain name is a registered Internet address that gets assigned to a static IP address that people use to connect to a web site. When a user enters a domain name into the address bar of a web browser, they are directed to the physical file that make up a web site. For this project, the domain name, "www.hwy34customs.com" was purchased, register and assigned a static IP address.

Virtual Domain Host

Virtual Domain Hosting is a necessary component of a "live" web site. The virtual domain host is a combination of hardware and software owned by an outside company (often but no always) on which the files that make up a web site physically reside. After using this particular virtual domain host for a variety of other web sites, the choice to host the Hwy 34 Customs web site with 1planhost was made. Among other advantages to 1planhost, if a domain name is register through 1planhost, 1planhost will pay all the annual domain name renewal fees as long as the domain name is hosted by 1planhost. One year of virtual domain hosting with 1planhost costs $99/year.

SQL Server 2000

The virtual domain host, "1planhost.com" provides 50Mb of SQL Server 2000 database for an additional $35/year over the regular hosting fee. However, they do not provide any of the tools used to operate SQL Server. Additionally, they only provide minimal technical support for the SQL Server database. A "warning" on their web site clearly states these facts and they recommend using an Access or My SQL database for those who are not familiar with SQL Server. SQL Server 2000 was chosen as the database for this project because it is a powerful database that allows for future expansion in a sound environment.

ASP.net

Microsoft Visual Studio.net provides a programming environment in which programmers can program in a variety of languages as well as create interactive Active Server Pages (ASP) web pages referred to as "web forms". One of the driving forces that make ASP.net function the way it does can be from the use of a background programming language such as C#, C++, or Visual Basic among others. This background programming language is know as "code behind". While some programmers prefer to use C# or other programming languages for their code behind, Visual Basic was chosen for this project. (Holzner, S. 2003)

III. Transact SQL

Stored Procedures

The Hwy 34 Customs web site is a dynamic, database driven web site, which uses a SQL Server 2000 database to store information about the inventory that administration at Hwy 34 Customs and collisions wish to display to the public via the Internet. SQL Server 2000 is a relational database that stores information called records in tables that are set up by a database administrator or perhaps a web page designer or other entity at that level. Tables and records are created, updated and deleted/dropped using a Standard Query Language (SQL). While there are a variety of versions of SQL such as Oracle's PL SQL, Microsoft's version known as Transact SQL (T-SQL) was used for this project. SQL sends commands and/or information to a database, which is used to create tables, update tables and delete tables among other things. This information can be hard-coded and sent straight to the database or it can be passed in through the use of parameters. This project utilizes a series of small programs written in SQL called stored procedures that collect information supplied through parameters. That information is then passed along to the database to alter the tables of the database. Records can be added to the tables, updated or permanently deleted from a table. The admin person at Hwy 34 Customs doesn't need to posses any working knowledge of SQL or any other programming language because the stored procedures work with the ASP.net pages to handle all the, "behind the scenes" work. Although a stored procedure can be named just about anything within SQL standards, this project utilizes a particular naming scheme, which incorporates some SQL logic as well and certain business rules. The "dbo" used in the stored procedure naming convention of this project specifies that the stored procedures belong to the DataBase Owner.

The reasoning behind the "spcc" is two-fold. It signifies to the programmer that this block of code is a stored procedure and it helps group the stored procedures in SQL Query Analyzer so the programmer doesn't have to search through long lists of files to find the stored procedure he wishes to work with.

dbo.spcc_AddCarsForSale

This stored procedure is used to add records to the CarsForSale table in the SQL Server 2000 database. The admin person will provide information by filling out text boxes on the AddCarsForSale.aspx page. When the "Submit" button is clicked, the information provided in the text boxes gets temporarily stored in parameters. These parameters are then passed into the stored procedure and used to add a new record to the table. The only field of the table that the admin person doesn't control is the FieldID field, which is automatically generated by the SQL Server 2000 database as an identity column to uniquely identify each individual records within that table. The FieldID remains attached to a particular record until that record is deleted from the table and is never re-used.

The stored procedure code to add a record to the CarsForSale page is as follows:

```
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
-- select * from CarsForSale (nolock)
/*************************************************
exec spcc_AddCarsForSale 'N/A', 'Old Blue', '1984', 'Chevy',
'Silverado', '$3500', '4x4'
*************************************************/
/*******************************************
This is to add records to the Cars For Sale page.
*******************************************/
ALTER    PROCEDURE dbo.spcc_AddCarsForSale
@tbCFSPicture              varchar(50),
@tbCFSName                 varchar(50),
@tbCFSYear                 varchar(4),
@tbCFSMake                 varchar(50),
@tbCFSModel                varchar(50),
@tbCFSPrice                varchar(8),
@tbCFSDescription          varchar(500)
AS
begin
INSERT INTO CarsForSale
(Picture, [Name], [Year], Make, Model, Price, [Description])
Values
(@tbCFSPicture, @tbCFSName, @tbCFSYear, @tbCFSMake,
@tbCFSModel, @tbCFSPrice, @tbCFSDescription)
end
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

Tables and Columns

The true backbone of the Hwy 34 Customs dynamic, database driven web site are the tables within SQL Server 2000. All the information about the inventory listed on the web site is stored in the tables that correspond to the inventory categories. In other words, all the information about "Cars For Sale" is stored in the CarsForSale table in the database. An admin person enters data (records) into the database by entering information in the text boxes on the ASP.net pages on the Admin Controls side of the web page and then clicks the "Submit" button. As a result, the text in the text boxes gets passed to stored procedures through parameters. The stored procedures use the parameters to enter information (records) into the database table. Each text box the admin person fills out represents a column (or field) in the database table. Screen shots of the actual tables in the db_hwy34Customs database as viewed in Enterprise Manager are located in Appendix A.

IV. Statement of Problem

Hwy 34 Customs is a relative new business to the Madison area. Dave Theilbar, owner of PSI Inc. built Hwy 34 Customs basically as a hobby with the idea of having fun building hot rods while making enough money to pay the bills and keep Hwy 34 Customs operational. As a savvy businessman and generally intelligent person with some working knowledge of computers and the Internet, Mr. Theilbar made the decision to advertise his new business products and services over the Internet.

Hwy 34 Customs needed a web site designed and built that would enable them to maintain the web site themselves without vast working knowledge of programming and Client Server technologies. They wanted a web site that worked similar to Cars.com in which they could simply log on and add, edit and delete their own products and services including pictures from a users perspective. To accomplish this, the following tasks had to be accomplished.

- Purchase a Domain Name
- Set up virtual domain hosting
- Set up a MS SQL Server 2000 database
- Create database tables
- Create a web site home page using frames in html
- Create ASP.net web forms to interacting with the database

V. Objectives

Deliverables

The Hwy 34 Customs web site uses a combination of several elements to form one basic deliverable, the web site itself. There are two basic parts of the web site, which area a result of many combined elements. There is the user or public side of the web site, which is what a user with an Internet connection can see when he browses the web site. The user has the ability to click on the navigation links on the left side of the web page and see lists of inventory for products and services provided by Hwy 34 Customs. The user will have the ability to read information about Hwy 34 Customs, find contact information, see information about stuff for sale, cars for sale, customer's cars, upholstery, current projects, future projects, the staff or even request a quote.

The other side of the web site deliverable is the administrative side or "Admin Controls". A user name and password is needed to access the "Admin Controls" side of the Hwy 34 Customs web site. With the proper user name and password, an admin person can log on to the "back side" of the web site where he can add information, edit information or even delete information from the inventory of the web site including pictures.

Components of the Project

Although there is one basic deliverable to this project, which is the Hwy 34 Customs web site that possesses two basic features, the users side and the admin side, there are many more "behind the scenes" components of the project that actually make it work.

The first component of the web site is the domain name "www.hwy34customs.com" which is an alias for a unique static IP address.

This IP address is register as Hwy 34 Custom's location on the Internet. Now, when someone types "www.hwy34customs.com" in the address bar of his favorite web browser, he will be automatically directed to the Hwy 34 Customs web site.

The next component of the web site is the virtual domain host. The virtual domain host is a combination of hardware and software owned by an outside company on which the web site physically resides. After using this particular virtual domain host for a variety of other web sites, the decision to host the Hwy 34 Customs web site with 1planhost was made. Among other advantages to 1planhost, if a domain name is register through 1planhost, 1planhost will pay all the annual domain name renewal fees as long as the domain name is hosted by 1planhost. One year of virtual domain hosting with 1planhost costs $99/year.

Another major component of the Hwy 34 Customs web site is the SQL Server 2000 database, which is an additional feature that can be ordered through 1planhost. 50mb of SQL Server 2000 database space can be added for an addition cost of $35/year. Once the SQL Server 2000 space is reserved, the actual database is set up and given a name. The Hwy 34 Customs database is named db_hwy34customs and consists of 7 working tables and several system tables. Other tables have been set up as well with expansion in mind but more tables can be added at any time as needed. The following user tables are currently being used on the Hwy 34 Customs web site.

- CarsForSale
- CurrentProject
- CustomersCars
- FutureProjects
- MeetTheStaff
- StuffForSale
- Upholstery

VI. Scope of Plan of Action

The Web Site

Hwy 34 Customs requires a database driven web site, which will give them the ability to administrate their own web content without working knowledge of programming and web site design or special software such as HTML editors or Cute FTP agents.

Since the web site is database driven, it made the most sense to work primarily at the database level. Although ASP.net can be used to directly alter data in a database, it was decided to combine ASP.net and Microsoft's version of SQL (T-SQL) to conduct the requested update anomalies.

SQL is the backbone of this project. Using ASP.net as the front end, users provide parameters that are passed to stored procedures, which in turn update the records in the database.

Layout and Design

The layout and design of Hwy 34 Custom's web site was developed to meet the requirements of the business as well as serve a purpose in the way the web site actually functions. The home page follows a simplistic frame design commonly used to keep a user within the working area of the main menu. Another reason for using frames is to avoid the repetition of re-loading navigation links and graphics on corresponding pages within the site. This web site uses three frames. There is a left frame for the link menu, a top-right frame for the Hwy 34 Customs logo and a bottom right frame, which is used as the main, working frame of the web site. The main frame is where the user will browse through the other pages of the web site while the left and top-right frames will never change.

The home page, which is named "index.htm" is actually a series of other HTML web pages called "source" pages. The left frame is really just an HTML page named, "LeftFrame", which is nothing more than the link menu.

The index page calls the LeftFrame.htm page and displays it on the left side of the web page. The top frame of the web site calls the TopFrame.htm page, which is simply the Hwy 34 Customs logo and displays it on at the top of the web page. When the page is first opened, the main frame calls the MainFrame.htm page, which is a picture designed specifically for this project and displays it in the main frame of the web site, which is where all the other pages of the web site will display and admin or the user navigate the site.

The Hwy 34 Customs web site is made up 4 basic HTML web pages and many aspx web pages, which were developed in MS Visual Studio.net. The following tables list each web page and type.

HTML Pages

HTML Pages

File	Extensions
Index	.htm
TopFrame	.htm
LeftFrame	.htm
MainFrame	.htm
AboutUs	.htm
ContactUs	.htm

ASP.net Pages

ASP.net Pages

File	Extension
AdminLogOn	.aspx
AdminControl	.aspx
AddCarsForSale	.aspx
AddCurrentProjects	.aspx
AddCustomersCars	.aspx
AddFutureProjects	.aspx
AddMeetTheStaff	.aspx
AddStuffForSale	.aspx
AddUpholstery	.aspx
ControlCarsForSale	.aspx
ControlCurrentProjects	.aspx
ControlCustomersCars	.aspx
ControlFutureProjects	.aspx
ControlMeetTheStaff	.aspx
ControlStuffForSale	.aspx
ControlUpholstery	.aspx
DeleteCarsForSale	.aspx
DeleteCurrentProjects	.aspx
DeleteCustomersCars	.aspx
DeleteFutureProjects	.aspx
DeleteMeetTheStaff	.aspx
DeleteStuffForSale	.aspx
DeleteUpholstery	.aspx
DisplayCarsForSale	.aspx
DisplayCurrentProjects	.aspx
DisplayCustomersCars	.aspx

File	Extension
DisplayFutureProjects	.aspx
DisplayMeetTheStaff	.aspx
DisplayStuffForSale	.aspx
DisplayUpholstery	.aspx
EditCarsForSale	.aspx
EditCurrentProjects	.aspx
EditCustomersCars	.aspx
EditFutureProjects	.aspx
EditMeetTheStaff	.aspx
EditStuffForSale	.aspx
EditUpholstery	.aspx
RequestQuote	.aspx

SQL Pages (Stored Procedures)

SQL Pages

File	File Type
dbo.spcc_AddCarsForSale	Stored Procedure
dbo.spcc_AddCurrentProjects	Stored Procedure
dbo.spcc_AddCustomersCars	Stored Procedure
dbo.spcc_AddFutureProjects	Stored Procedure
dbo.spcc_AddMeetTheStaff	Stored Procedure
dbo.spcc_AddStuffForSale	Stored Procedure
dbo.spcc_AddUpholstery	Stored Procedure
dbo.spcc_DeleteCarsForSale	Stored Procedure
dbo.spcc_DeleteCurrentProjects	Stored Procedure
dbo.spcc_DeleteCustomersCars	Stored Procedure
dbo.spcc_DeleteFutureProjects	Stored Procedure
dbo.spcc_DeleteMeetTheStaff	Stored Procedure

File	File Type
dbo.spcc_DeleteStuffForSale	Stored Procedure
dbo.spcc_EditCarsForSale	Stored Procedure
dbo.spcc_EditCurrentProjects	Stored Procedure
dbo.spcc_EditCustomersCars	Stored Procedure
dbo.spcc_EditFutureProjects	Stored Procedure
dbo.spcc_EditMeetTheStaff	Stored Procedure
dbo.spcc_EditStuffForSale	Stored Procedure
dbo.spcc_EditUpholstery	Stored Procedure
dbo.spcc_DisplayCarsForSale	Stored Procedure
dbo.spcc_DisplayCurrentProjects	Stored Procedure
dbo.spcc_DisplayCustomersCars	Stored Procedure
dbo.spcc_DisplayFutureProjects	Stored Procedure
dbo.spcc_DisplayMeetTheStaff	Stored Procedure
dbo.spcc_DisplayStuffForSale	Stored Procedure
dbo.spcc_DisplayUpholstery	Stored Procedure
dbo.spcc_DisplayCarsForSaleInTextbox	Stored Procedure
dbo.spcc_DisplayCurrentProjectsInTextbox	Stored Procedure
dbo.spcc_DisplayCustomersCarsInTextbox	Stored Procedure
dbo.spcc_DisplayFutureProjectsInTextbox	Stored Procedure
dbo.spcc_DisplayMeetTheStaffInTextbox	Stored Procedure
dbo.spcc_DisplayStuffForSaleInTextbox	Stored Procedure
dbo.spcc_DisplayUpholsteryInTextbox	Stored Procedure

SQL Tables

SQL Tables

Table Name	Database
CarsForSale	db_hwy34Customs
CurrentProjects	db_hwy34Customs
StuffForSale	db_hwy34Customs
CustomersCars	db_hwy34Customs
FutureProjects	db_hwy34Customs
MeetTheStaff	db_hwy34Customs
Upholsterys	db_hwy34Customs

VII. Functionality

How it Works

There are several working components to the Hwy 34 Customs web site that work together to create the functionality of the web site from both the administrative and user's perspectives. What a user can see and do while visiting the web site differs from what someone with administrative rights can do. A user can browse the web site and see a brief description and picture (if applicable) of all the inventory Hwy 34 Customs is currently advertising. Someone with administrative rights however can log on to the "Admin Controls" side of the web site with a user name and password to add, edit and delete items/ records from the web site's database.

Information about each product is stored in the corresponding tables of the database. For example, one table in the database stores information about "Current Projects" as demonstrated in Example Table 1, where Field ID is the Identity field.

Example Table 1

Field ID	Picture	Name	Year	Make	Model	Description
1	BlueFirebird.jpg	Dave S.	1967	Pontiac	Firebird	Blue
2	59Chevy.jpg	Bill D.	1959	Chevy	Apache	Black
3	T-Top.jpg	John C.	1980	Chevy	Camaro	White

Pictures are not actually stored in the database, they are stored in a directory named "ImgDir", which resides on the virtual domain server. The path to the picture in that directory is stored in the database.

Anyone with administrative rights can log on to the web site's Admin Controls and add records, edit records and delete records from the database as well as upload images. For example, if the white "T-Top" represented by the Field ID 3 gets a new paint job, the description would change to the new color as demonstrated in Example Table 2:

Example Table 2

Field ID	Picture	Name	Year	Make	Model	Description
1	BlueFirebird.jpg	Dave S.	1967	Pontiac	Firebird	Blue
2	59Chevy.jpg	Bill D.	1959	Chevy	Apache	Black
3	T-Top.jpg	John C.	1980	Chevy	Camaro	Orange

Let's examine the process of how that change was made, starting with how the records was created and displayed in the first place.

Scenario of Adding / Editing / Deleting records

Dave Thielbar, the owner of Hwy 34 Customs accepted the job of painting John C.'s white 1980 Chevy T-Top. Dave decided that this project would make a nice addition to the Hwy 34 Customs web site so he logs on to the Admin Controls side of the web site, clicks the "Current Projects Link" and then clicks the "Add Current Projects" link located at the top of the page. From there, Dave is taken to the AddCurrentProjects.aspx page where he fills out a form, supplying all the information as stated in table (5). When the "Submit" button is clicked, the new information gets sent to a stored procedure through parameters. The stored procedure uses the information supplied by the parameters from the ASP.net page to add a new record to the database.

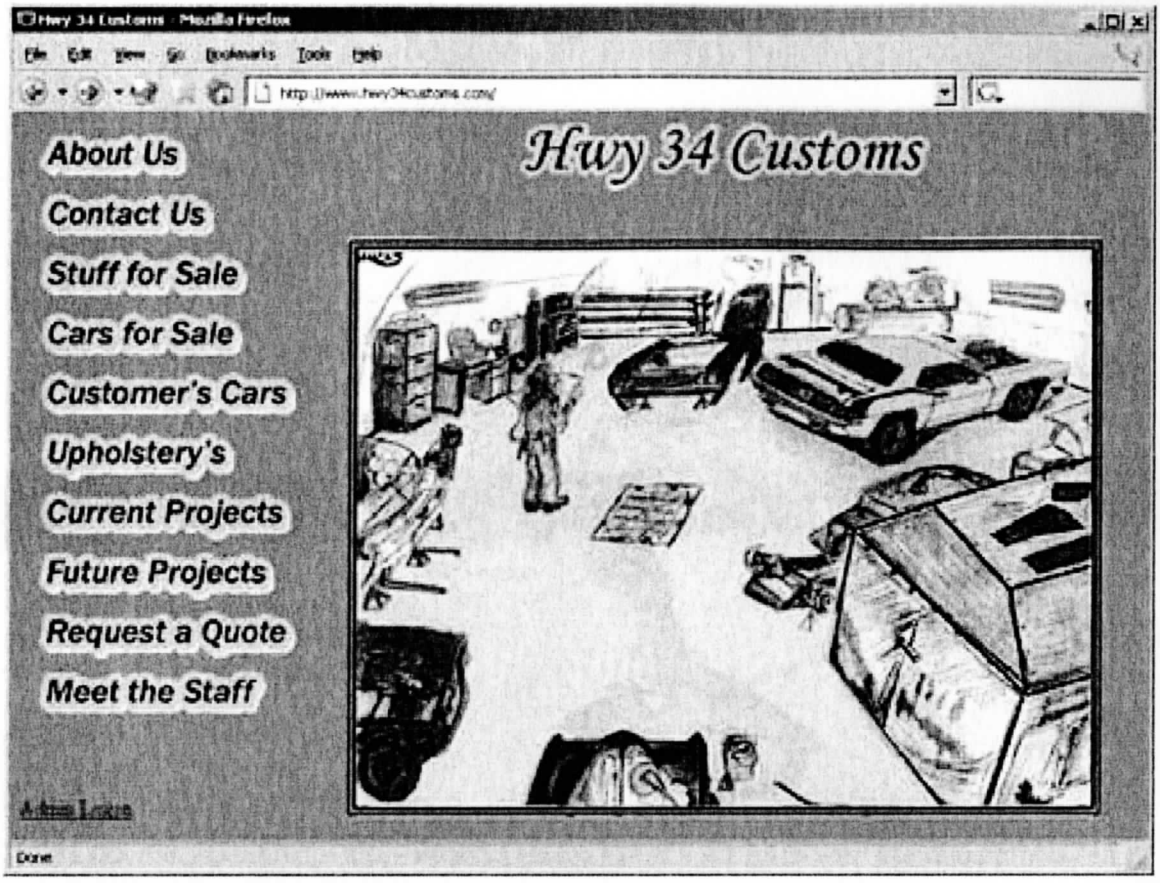
The same basic principle is used to edit records within the database. Parameters send information to a stored procedure, which update records with the new information supplied by admin. Records are deleted by using the same process using only the FieldID field where a record is deleted by FieldID.

What Each Component Does

HTML page are used to display static information on the Hwy 34 Customs web site as well as to establish a working environment for the dynamic ASP.net web pages. Tables on MS SQL Server 2000 are used to store information/records about each inventory item. ASP.net pages are used as a front end for admin to supply information, which gets passed to stored procedures through the use of parameters. Other ASP.net pages are used to retrieve information from the SQL Server 2000 database and display that information to users. Stored procedures, which reside on SQL Server 2000 collect information from the parameters supplied by the ASP.net pages and use those parameters to add, edit and delete records in the database.

Index.htm

Index.htm



The Index.htm page doesn't display anything on it's own, it simply acts as a frameset that is used to display other html and aspx web pages. Image 1 shows the Index frameset and as well as three other HTML pages that provide the actual content of the page. The content of the page is displayed using the following HTML web pages:

- TopFrame.htm
- LeftFrame.htm
- MainFrame.htm

The TopFrame.htm page only contains the graphic that says "Hwy 34 Customs. The LeftFrame.htm page contains the link to "Admin Log on" as well as the rollover links that say:

- About Us
- Contact Us
- Stuff For Sale
- Cars For Sale
- Customers Cars
- Upholsterys
- Current Projects
- Future Projects
- Request a Quote
- Meet The Staff

The MainFrame.htm page contains the graphic depiction of a shop, which is used as the home page of the web site. The "MainFrame" frame is also where the rest of the web site will be displayed as the user or admin moves around within the web site.

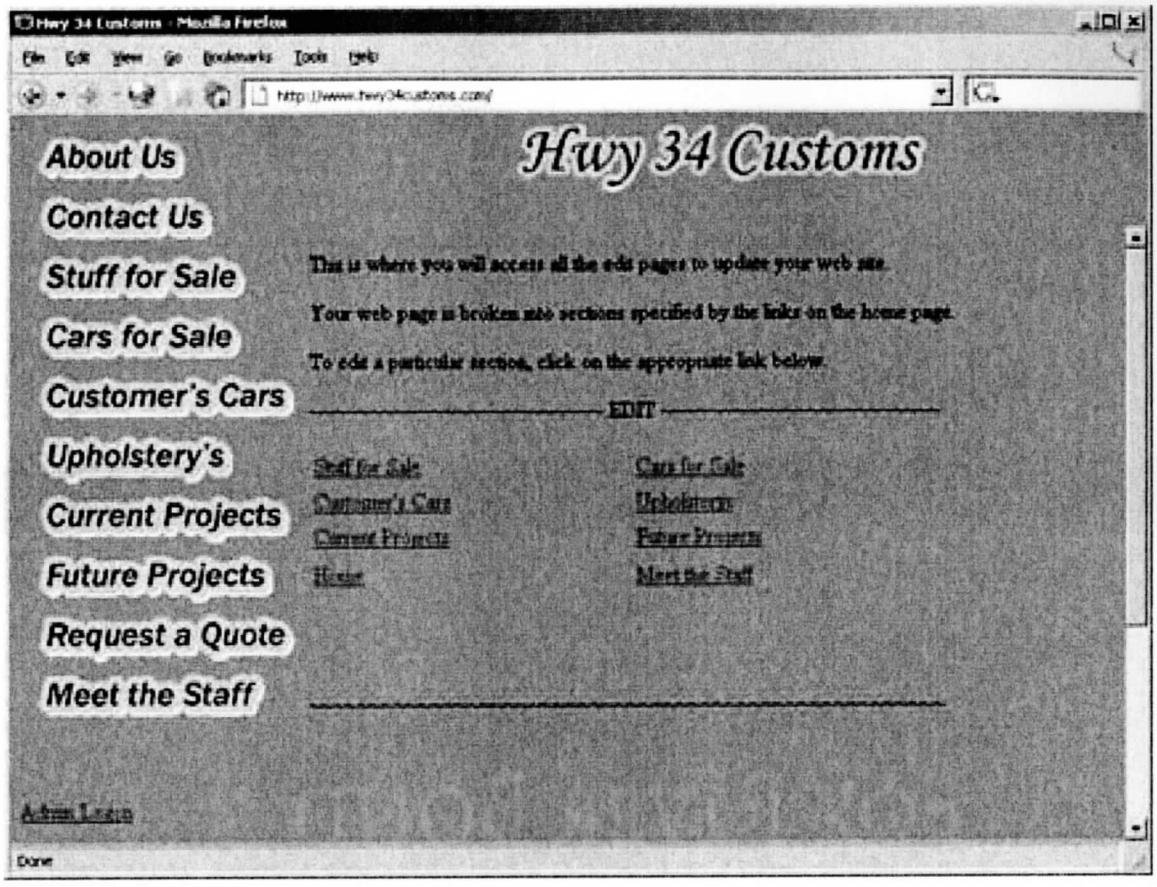
AdminLogOn.aspx

AdminLogOn.aspx

The AdminLogOn.aspx page only serves one purpose, which is a segway into the heart of the web site. An administrator will enter his user name and password (which were supplied) into the text boxes and click the "Log On" button. If the correct user name and password are entered, the administrator will be taken to the Admin Controls page (AdminControls.aspx). Included on the AdminLogOn.aspx page are form validators, which will return error messages if an incorrect user name or password were entered and no user name or password were entered.

AdminControls.aspx

AdminControls.aspx

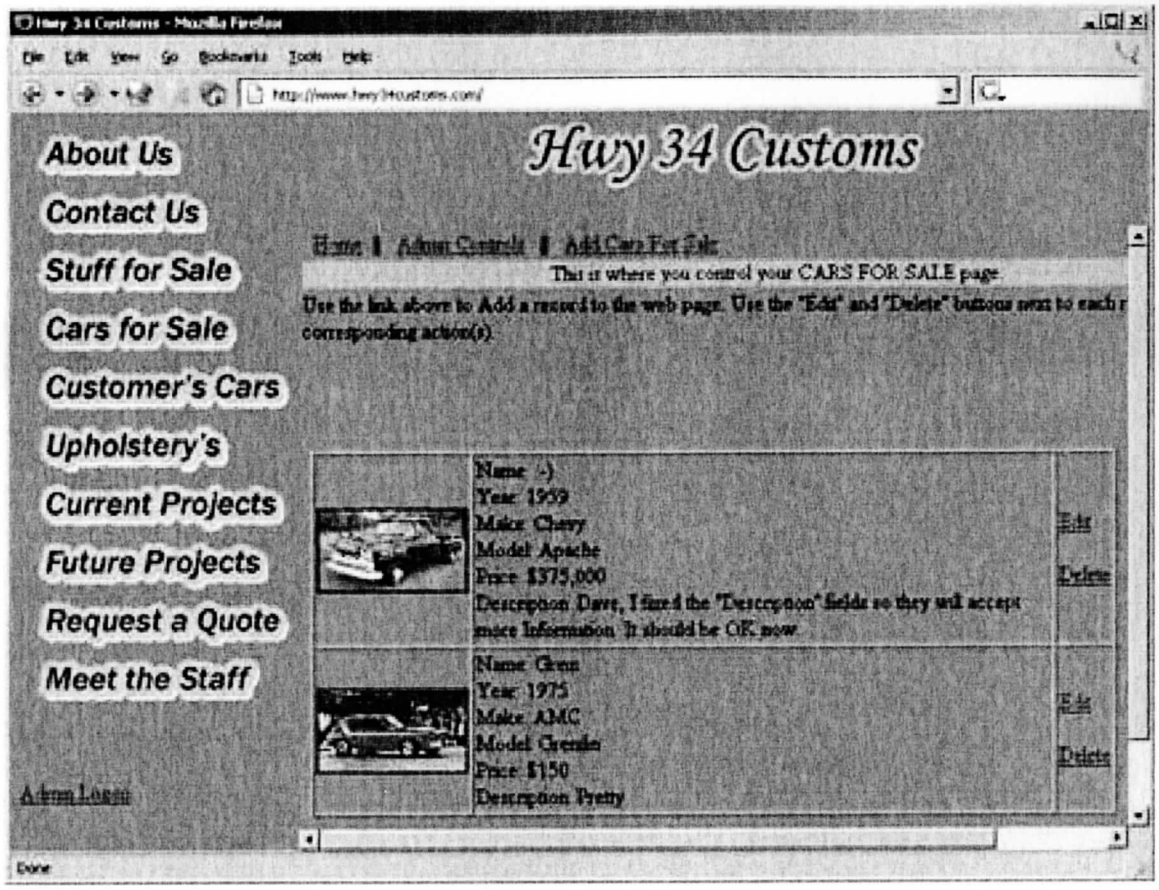


The AdminControls.aspx page functions as a portal to the real working pages of the web site. Image 3 shows how the Index.htm page uses frames to display page information. As with the AdminLogOn.aspx page (and all the other ASP.net pages) the AdminControls.aspx page is displayed in the "MainFrame" (lower right corner) of the web page leaving the Hwy 34 Customs logo at the top of the page and the navigation links on the left side of the page. From the Admin Controls page, a person with administrative rights can navigate to the individual working pages of the web site that control each area of the web site.

ControlCarsForSale.aspx

By following the "Cars for Sale" link from the Admin Controls controls page, the admin person is taken to the ControlCarsForSale.aspx page.

ControlsCarsForSale.aspx



This page is another segway that digs deeper into the actual working pages of the web site. At this point however, the admin person is dealing with an area of the web site that is specific to "Cars For Sale". Links from this page will deal directly to the Cars for Sale area of the database as well as effect the Cars for Sale areas of the web site itself.

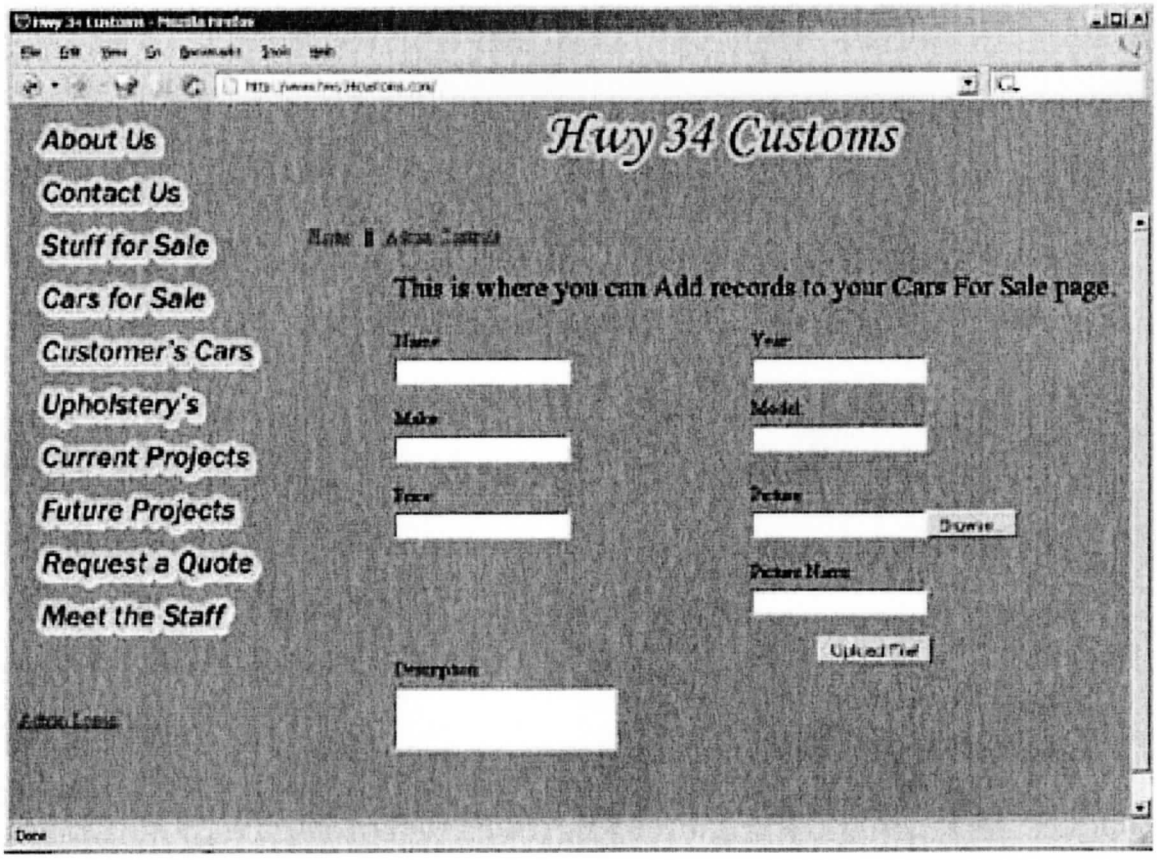
From this page, the admin person can navigate back to the Home page of the web site, back to the Admin Controls page to edit other areas of the database and web site, to the AddCarsForSale.aspx page where he can add records to the inventory or the EditCarsForSale.aspx page where he can edit existing records.

There is also an option from this location where admin can delete records by simply clicking the "Delete" link associated with a specific item.

AddCarsForSale.aspx

As mentioned above and displayed in Image 4, an admin person will be directed to the Add Cars For Sale page when he clicks the Add Cars For Sale link from the ControlCarsForSale.aspx page.

AddCarsForSale.aspx



This is the page that an administrator of the web page can add cars for sale to the inventory of the web site. What he is actually doing at this level is adding a record to the SQL Server database as well as uploading a picture to the "ImgDir" mentioned earlier.

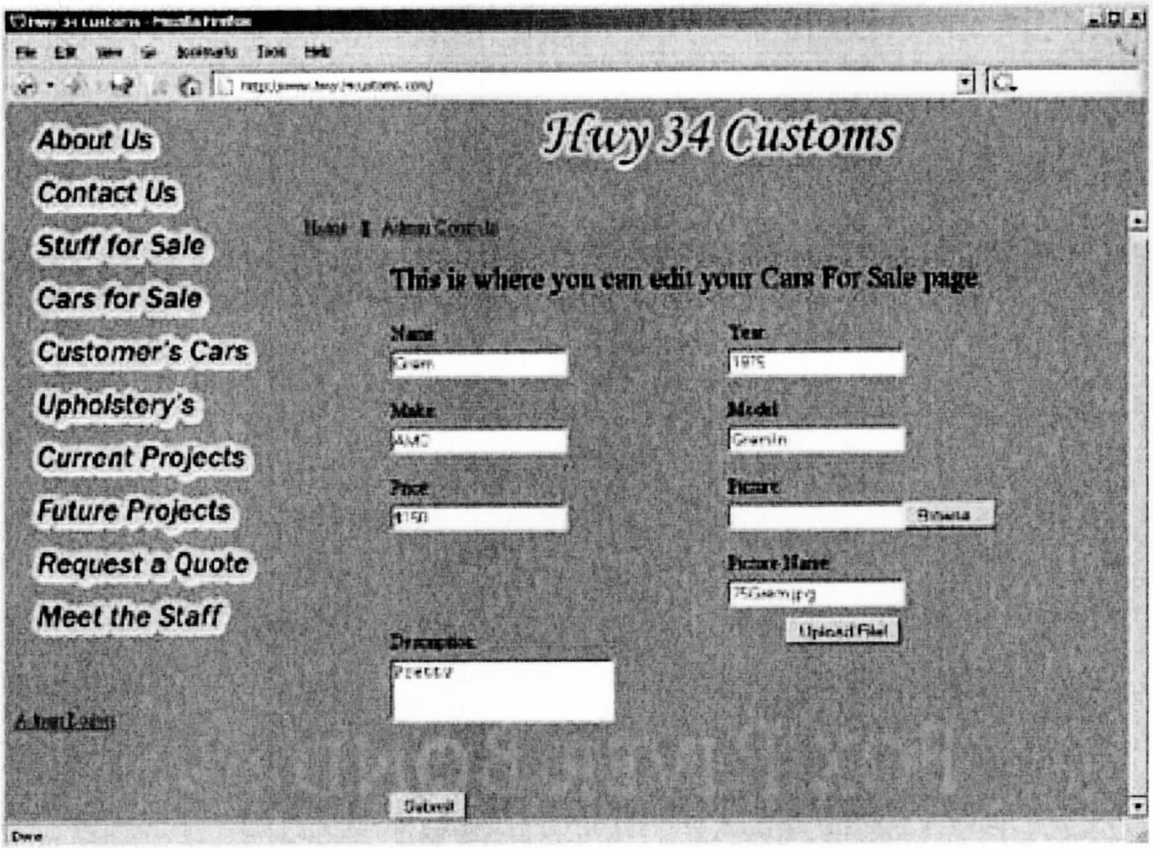
Each text box represents a field in the database.

When an administrator enters text into one of the text boxes and clicks the "Submit" button (not shown), the value/text gets stored in a parameter, which gets passed to the stored procedure dbo.spcc_AddCarsForSale. That stored procedure will then use that parameter information to create a new record in the "CarsForSale" database table. From the AddCarsForSale.aspx page, the admin person also has the ability to upload a picture associated with the inventory item. This is a simple three-step process from the admin person's perspective. To add a picture, the admin person simply clicks the "Browse" button, where he is prompted to search a local directory and select a picture to upload. This is the standard Windows process. Once a picture is selected for upload, the admin person must enter the corresponding name of the picture followed by the correct file extension (typically .jpg). For example, if the admin person select to upload the picture "BlueFirebird.jpg" from the C:\My Documents\My Pictures directory, he would enter BlueFirebird.jpg in the "Picture Name:" text box. At this point, he simply clicks the "Upload File!" button. The picture is then uploaded to the ImgDir directory on the virtual domain host. The name, "BlueFirebird.jpg" is stored in the CarsForSale table of the database in the "picture" field.

EditCarsForSale.aspx

Now that the database contains a new record in the "CarsForSale" database table, the admin person will have the ability to make modifications to that records using the EditCarsForSale.aspx page.

EditCarsForSale.aspx



The EditCarsForSale.aspx page is accessed by clicking the Edit link that is associated with the individual record on the ControlCarsForSale.aspx page. When the link is clicked, the Edit page identifies the corresponding record by a hidden identity field in the database table named "FieldID". The information contained in the database for that record is then pulled from the database and put in the text boxes on the Edit page.

This will prevent the admin person from having to remember the fields associated with record he is attempting to update as the text boxes will automatically be filled with the current information.

Identity Field

The identity field is a unique identifier that uniquely identifies each individual record in the database table. No two records within the same table can share the same identity field (FieldID in this project). By using the identity field to dynamically identify records within the database tables, the admin person is supplied with the ability to edit records without having to remember any particulars. If he wishes to edit a particular record, he simply clicks the "Edit" link associated with that record and the ASP.net page will remember the identity field (FieldID) and edit the corresponding record accordingly.

As with the AddCarsForSale.aspx page, the EditCarsForSale.aspx page uses text boxes in which the admin person supplies parameters by entering text into the text boxes. When the "Submit" button is clicked, the text within the text boxes gets stored in parameters and passed to the stored procedure dbo.spcc_EditCarsForSale. Using the parameters supplied from the text in the text boxes, this stored procedure will update the record that corresponds with the hidden identity field, (FieldID), which cannot be changed. Only new records will get new field ids.

DeleteCarsForSale.aspx

There isn't a screen shot of this page because no one ever sees this page. If anyone were to see this page through a web browser, all he would see is a blank screen as there are no visible parts to this ASP.net page. Although there are no visible parts to

DeleteCarsForSale.aspx, it still serves a purpose and has functionality. The DeleteCarsForSale page is activated from the EditCarsForSale.aspx page.

It is an intermediate page that performs a specific task and then instantly redirects the admin person back to the EditCarsForSale.aspx page. Rather than collect information in a text box like the other ASP.net pages, the DeleteCarsForSale.aspx page merely accepts the FieldID parameter from the EditCarsForSale.aspx page, which it then passes to the stored procedure dbo.spcc_DeleteCarsForSale, which uses that parameter to delete the corresponding record from the CarsForSale table in the SQL Server 2000 database. As soon as the parameter is passed through the DeleteCarsForSale.aspx page into the stored procedure and the record is deleted from the database, the DeleteCarsForSale.aspx page redirects the admin person back to the EditCarsForSale.aspx where the delete record is no longer present. The process appears seamless.

VIII. Results / Conclusion

The Hwy 34 Customs web site was successfully designed and built with minimal complications. The most difficult part of the project was making the ASP.net pages work on a remote server. Coding the ASP.net pages wasn't that difficult but through all the course work in the MSIS program, the opportunity to implement files on a remote server was never came up. Only localhost debugging on the local server occurred. After discovering how to use .dll files and what host directory to upload them to, the project was up and running.

The design and technical approach to building this web site is efficient, logical and somewhat simplistic in implementation. Although there are many pages of code that work together to bring the web site together as a working whole, none of the code by itself is very complex.

Future Plans for www.hyw34Customs.com

One of the objectives of the Hwy 34 Customs web site was to design and build a structure that could expand over the course of time as the business changes and grows. Since this is a dynamic, database driven web site, the design is structured around a MS SQL Server 2000 database, which can change and expand with the business. There is currently 50mb of SQL Server 2000 database space allocated to www.hwy34customs.com for an annual fee of $35 but an additional 50mb of space can be added for another $35/year. Since the inventory pictures are being stored in a separate (non-database) directory on the host server, the only thing stored in db_hwy34customs is text. A lot of text can be stored in 50mb so the probability of needing more than 100mb of database space is relatively low.

Additional Features

Although the Hwy 34 Customs web site already exceeds the criteria from the project proposal by more than 100%, there are additional features planned for the web site. Some of these features are already started and sitting in wait. Some of the additional features include:

- Add/Edit/Delete Users
- Edit Contact Information
- Edit About Us Information

Add/Edit/Delete Users

Future plans for the Hwy 34 Customs web site include the ability for admin to add, edit or delete users who can log on to the web site's, "Admin Controls". In this situation, someone who is already logged on to the web site's, "Admin Controls" will be able to add another user name and password so additional people can have access to the "Admin Control" using their own unique identification. This information will be stored and accessed in a database table named, "Users", which is already created and laying in wait.

Edit Contact Information

Although it's not that likely, there is a future possibility that some of Hwy 34 Custom's contact information will change over the course of time. Perhaps the business will change telephone numbers and/or get a new fax number or email address. With the future additions coming to the web site, anyone with access to "Admin Controls" will have the ability to edit contact information in the same way they currently edit inventory information. Once again, this information will be stored in a database table, which is already created and named ,"ContactInfo".

The current contact information is displayed to the user via a standard html web page. A new ASP.net web page will be created to interact with the new database table to display the contact information to the user.

Edit About Us

As with the contact information, it's fairly unlikely that the "About Us" information will change very often. However, future plans for the Hwy 34 Customs web site include the addition of another ASP.net web page and database table that will be used to give the ability of an admin person to edit the "About Us" information on the web page in the same way he can edit the other information.

Enhancements to existing pages

Hwy 34 Customs offers an extensive range of products and services, which is advertised on the Internet via their web site. Some of the inventory exists as a whole or finished products such as "Cars For Sale" and some of the "Stuff For Sale". Most of the services provided by Hwy 34 Customs however expand over the course of progression. A "Current Project" for example might start out as a rusty old junker but end up as a prize winning show car. One of the first enhancements made to the Hwy 34 Customs web site will be the ability to add several pictures associated to inventory items so the user will have an opportunity to see a project over the course of time. There would be pictures of the car/ project when it first arrives at Hwy 34 Customs, pictures of the car/project when it's finished and pictures of the car/project at various stages of the build. Although the work on this enhancement hasn't been started yet, the design ideas have.

Basically, there will additional pages added, which will enable the current display and edit pages to act as an intermediary. From a user's perspective, he will see a list of the current inventory in the same way its currently displayed. With the future enhancements however, when the user clicks on the thumbnail to see a larger view of the picture, he will be taken to a "Detail" page in which there will be additional information about the inventory item along with several additional thumbnails that will link to full size images.

From an admin person's perspective, he will merely have more image upload options when adding or editing an inventory item from "Admin Controls". One possibility being considered is to simply have a limited number of image upload text boxes available from the current Add or Edit pages. Another possibility being considered is to have an additional page associated to each inventory item specifically for adding more pictures. Further research and discussion is required before a final decision is made.

Conclusion

The implementation of the Hwy 34 Customs web site was completed on schedule with minimal complication. This dynamic, database driven web site provides a user-friendly interface for users and admin to complete their desired tasks. The project was successful

IX. References

Bennet Wm. McEwan, David Solomon (1997). *Teach Yourself Transact-SQL in 21 days*: Sams Publishing

Holzner, S. (2003). *Teach Yourself Microsoft Visual Basic .NET 2003 in 21 Days*: Sams Publishing.

Microsoft asp.net *Accessing SQL-based Data* Retrieved March 12, 2005 from, http:// asp.net/Tutorials/quickstart.aspx

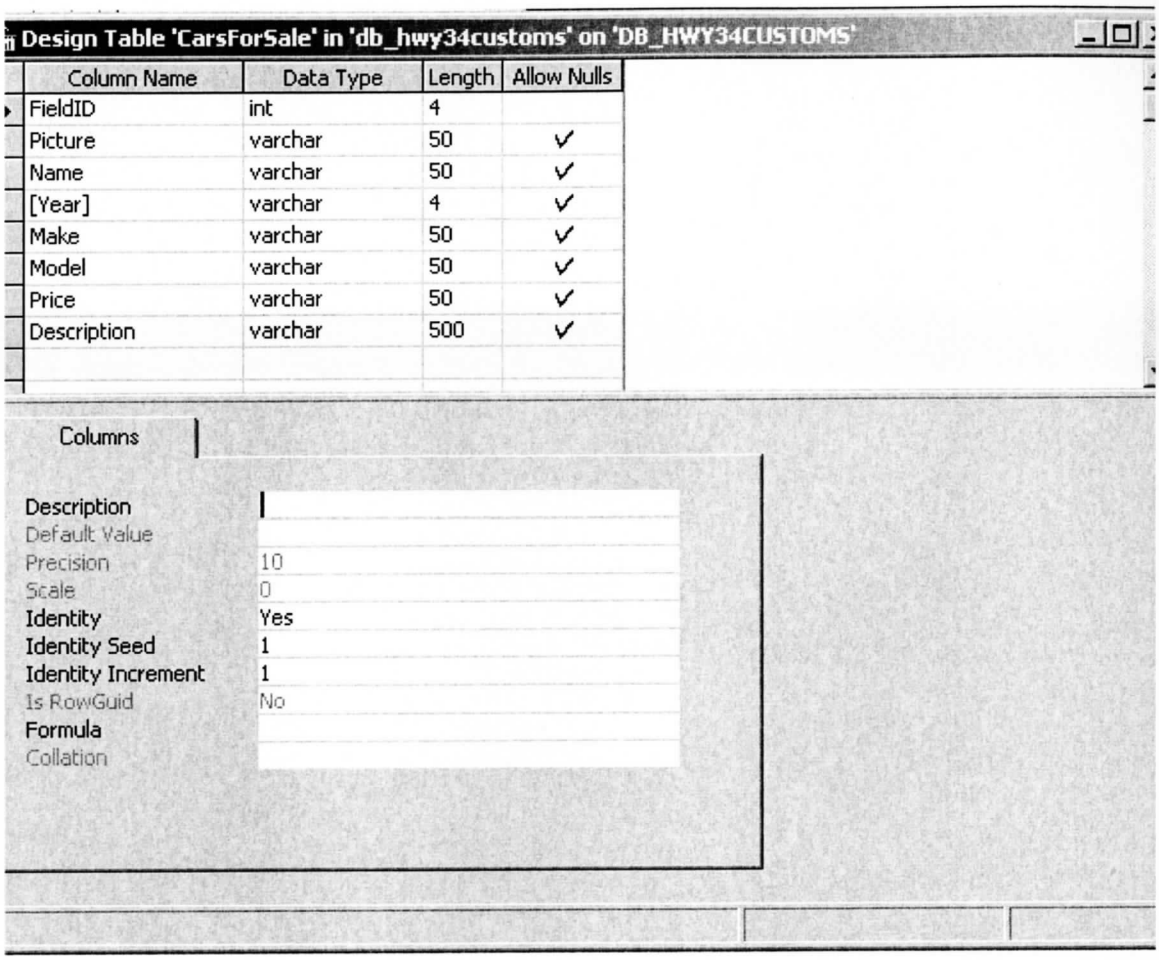
Microsoft asp.net *Writing and Using Stored Procedures* Retrieved March 10, 2005 from http://asp.net/Tutorials/quickstart.aspx

Walther, S. (2004). *ASP.NET Unleashed* (Second ed.): Sams Publishing.

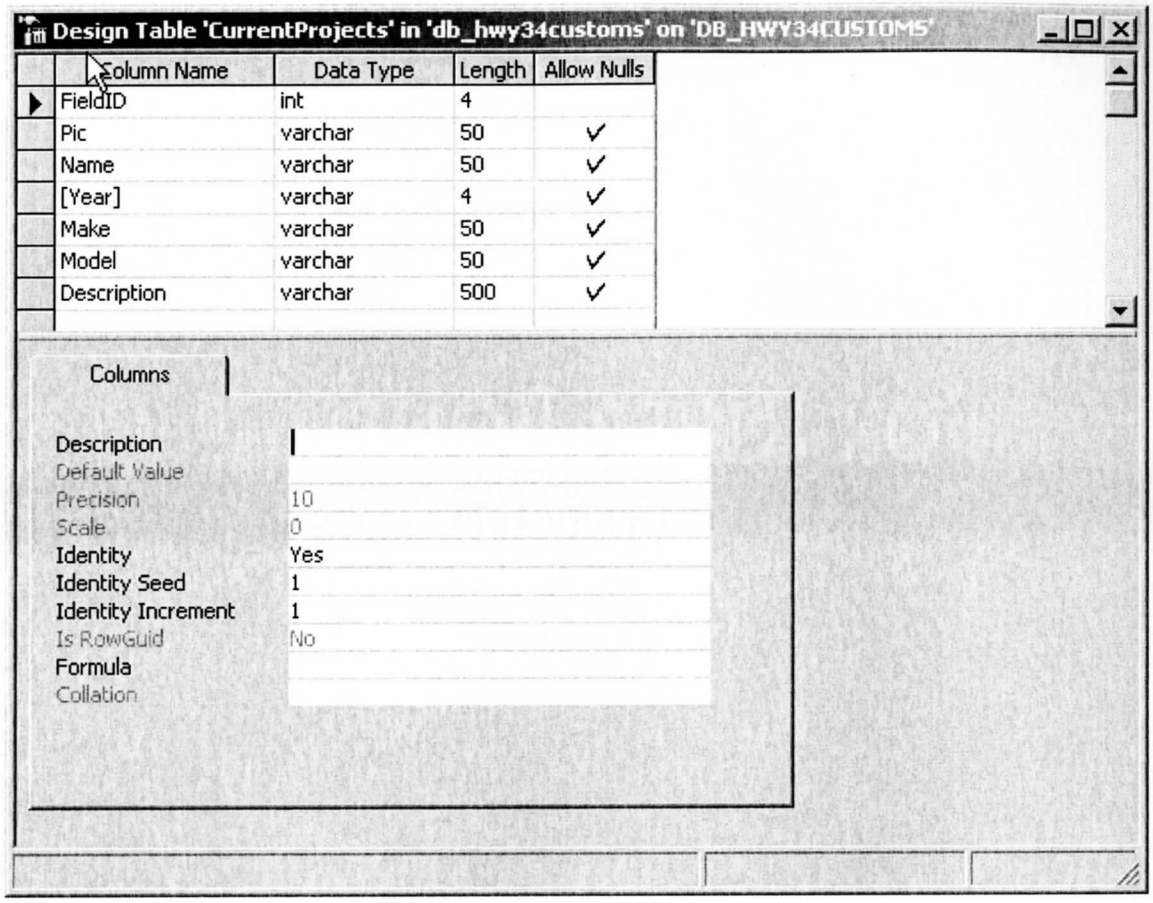
Appendix A

SQL Server 200 Tables

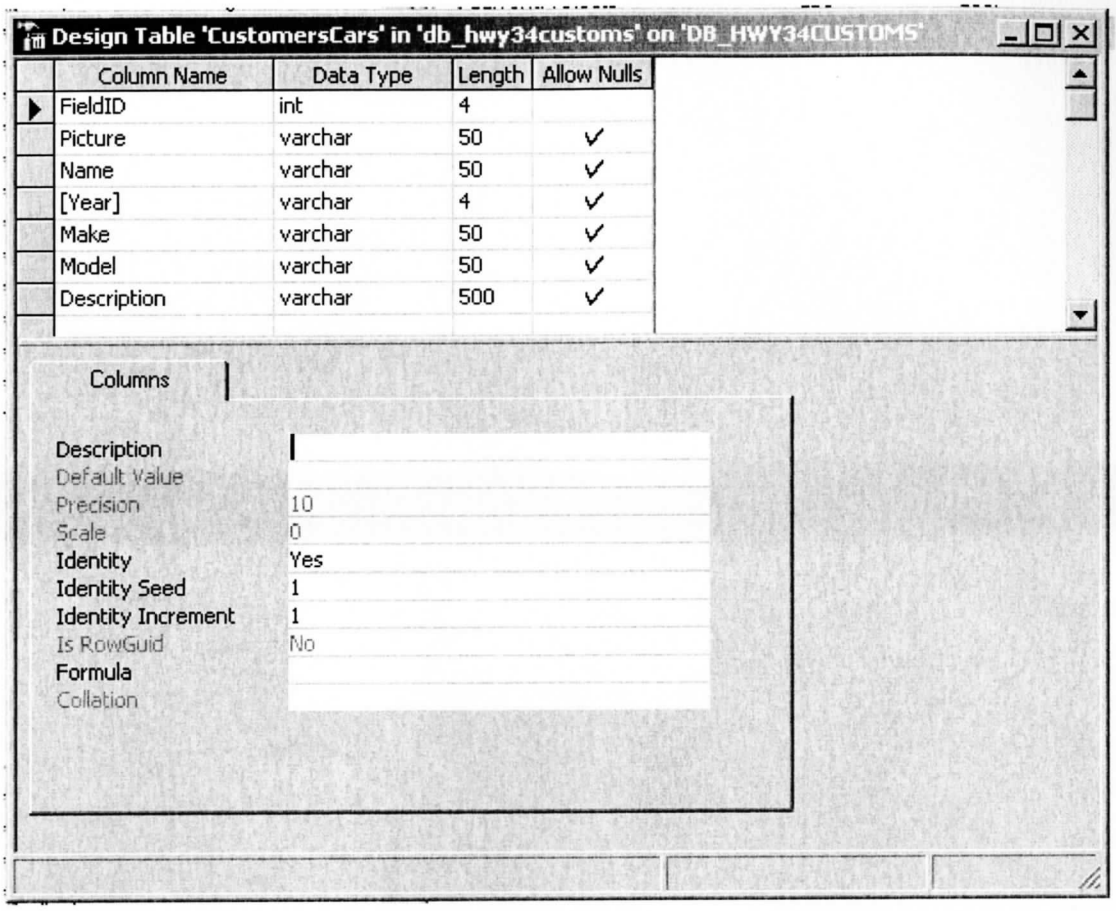
CarsForSale



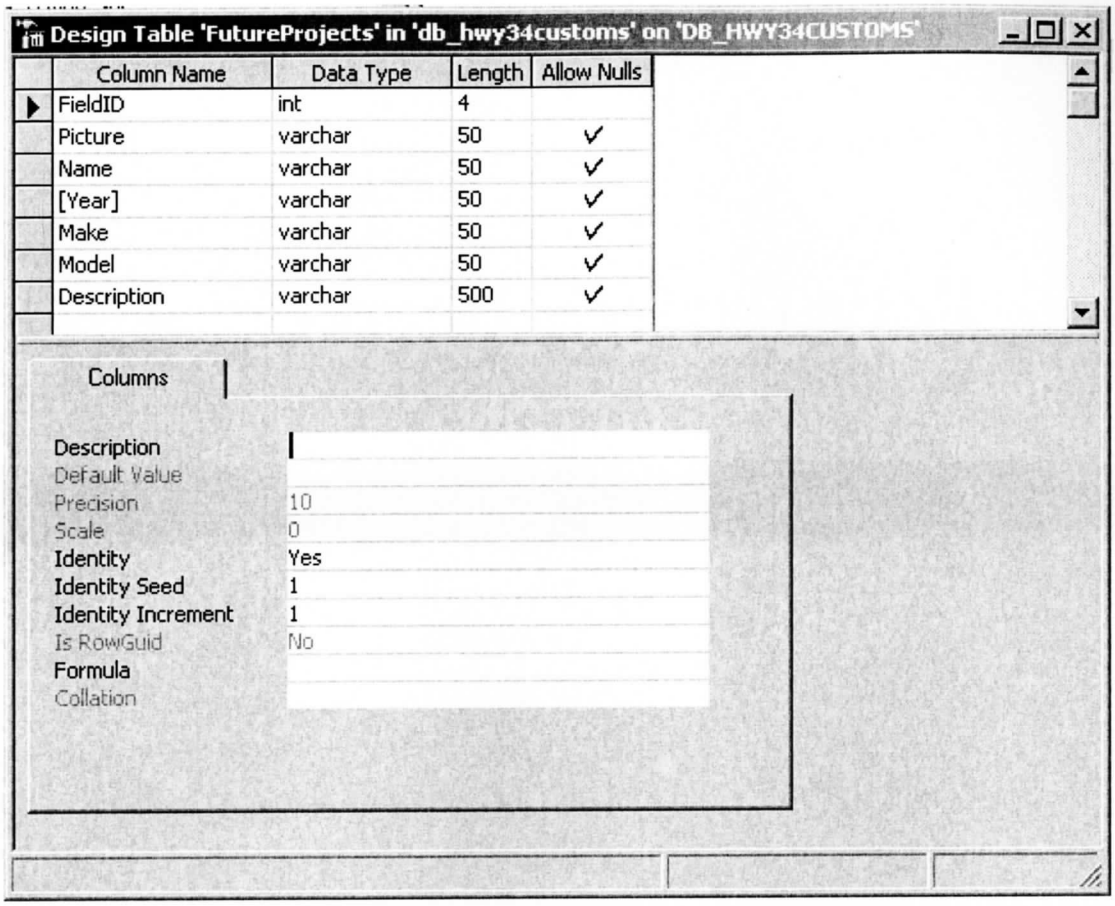
CurrentProjects



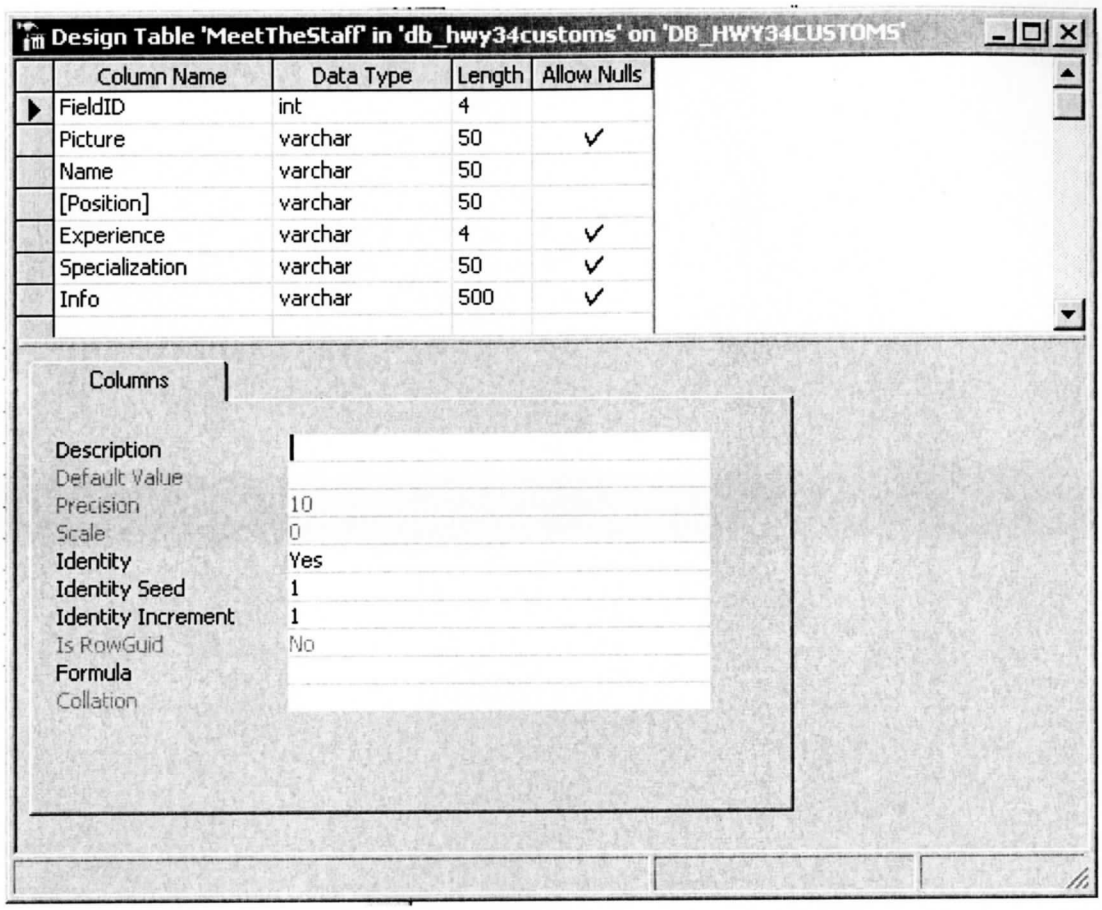
CustomersCars



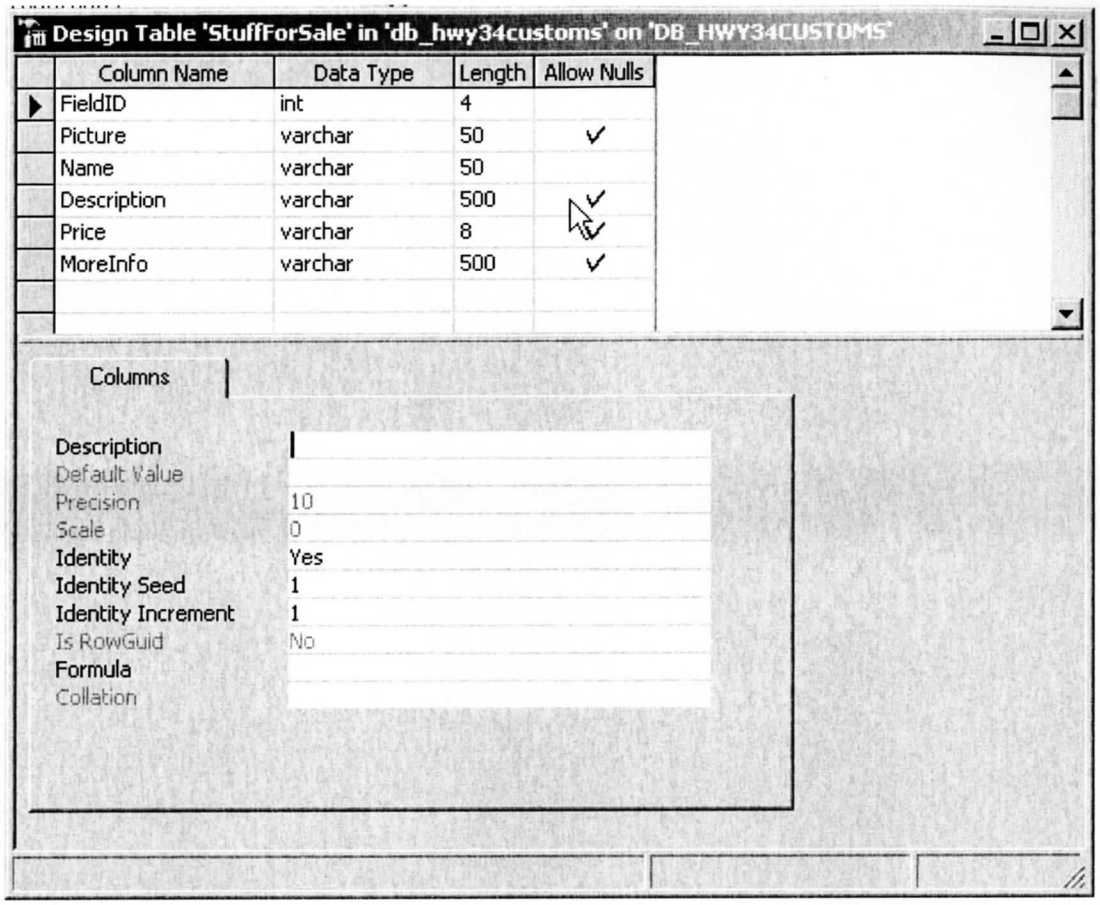
FutureProjects



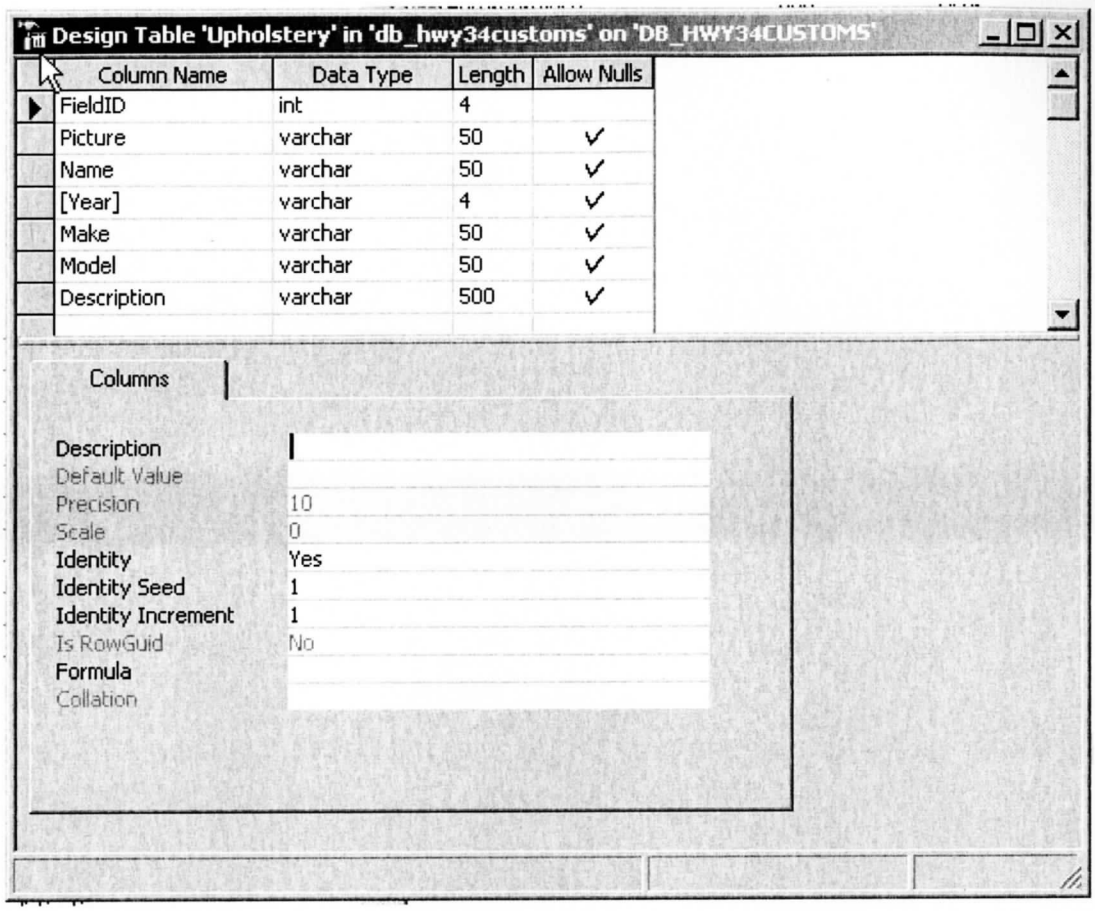
MeetTheStaff



StuffForSale



Upholstery


Design Table 'Upholstery' in 'db_hwy34customs' on 'DB_HWY34CUSTOMS'
Column Name
Data Type
Length
Allow Nulls
FieldID
int
4
Picture
varchar
50
Name
varchar
50
[Year]
varchar
4
Make
varchar
50
Model
varchar
50
Description
varchar
500
Columns
Description
Default Value
Precision
10
Scale
0
Identity
Yes
Identity Seed
1
Identity Increment
1
Is RowGuid
No
Formula
Collation

Appendix B

Sample (VB) Code Behind

There are several areas of the Hwy 34 Customs web site that work similarly from a technical aspect as well as from a users perspective. The programming and functionality for the "Cars for Sale" section web site uses the same principles and technology as the other areas of the web site even though the actual content varies. For this reason, only the basic admin controls and "Cars For Sale" related code is included in this appendix.

AdminControls.aspx

```
Public Class AdminControls
   Inherits System.Web.UI.Page

#Region " Web Form Designer Generated Code "

   'This call is required by the Web Form Designer.
   <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

   End Sub
   Protected WithEvents HyperLink1 As System.Web.UI.WebControls.HyperLink
   Protected WithEvents lnkCarsForSaleControl As
System.Web.UI.WebControls.HyperLink
   Protected WithEvents lnkMeetTheStaffControl As
System.Web.UI.WebControls.HyperLink
   Protected WithEvents lnkCCControl As System.Web.UI.WebControls.HyperLink
   Protected WithEvents lnkCPControl As System.Web.UI.WebControls.HyperLink
   Protected WithEvents lnkFPControl As System.Web.UI.WebControls.HyperLink
   Protected WithEvents lnkUpholsterysControl As
System.Web.UI.WebControls.HyperLink
   Protected WithEvents lnkSFSConrtols As System.Web.UI.WebControls.HyperLink

   'NOTE: The following placeholder declaration is required by the Web Form Designer.
   'Do not delete or move it.
   Private designerPlaceholderDeclaration As System.Object

   Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
      'CODEGEN: This method call is required by the Web Form Designer
      'Do not modify it using the code editor.
      InitializeComponent()
   End Sub

#End Region
```

```
    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Put user code to initialize the page here
    End Sub

End Class
```

AddCarsForSale.aspx

```
Imports sc = System.Data.SqlClient
Public Class AddCarsForSale
    Inherits System.Web.UI.Page
#Region " Web Form Designer Generated Code "
    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
    End Sub
    Protected WithEvents tbCFSDescription As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label5 As System.Web.UI.WebControls.Label
    Protected WithEvents tbCFSModel As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label4 As System.Web.UI.WebControls.Label
    Protected WithEvents tbCFSMake As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label3 As System.Web.UI.WebControls.Label
    Protected WithEvents tbCFSYear As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label2 As System.Web.UI.WebControls.Label
    Protected WithEvents Label6 As System.Web.UI.WebControls.Label
    Protected WithEvents HyperLink2 As System.Web.UI.WebControls.HyperLink
    Protected WithEvents Label7 As System.Web.UI.WebControls.Label
    Protected WithEvents HyperLink3 As System.Web.UI.WebControls.HyperLink
    Protected WithEvents tbCFSPrice As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
    Protected WithEvents Label8 As System.Web.UI.WebControls.Label
    Protected WithEvents btnAddCFS As System.Web.UI.WebControls.Button
    Protected WithEvents Label9 As System.Web.UI.WebControls.Label
    Protected WithEvents tbCFSName As System.Web.UI.WebControls.TextBox
    Protected WithEvents lblComplete As System.Web.UI.WebControls.Label
    Protected WithEvents Button1 As System.Web.UI.WebControls.Button
    Protected WithEvents Label10 As System.Web.UI.WebControls.Label
    Protected WithEvents inpFileUp As System.Web.UI.HtmlControls.HtmlInputFile
    Protected WithEvents tbCFSPicture As System.Web.UI.WebControls.TextBox
    'NOTE: The following placeholder declaration is required by the Web Form Designer.
    'Do not delete or move it.
    Private designerPlaceholderDeclaration As System.Object
    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
```

```
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub
#End Region
    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Put user code to initialize the page here
    End Sub

    Private Sub btnAddCFS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddCFS.Click
        Dim cnSQL As New sc.SqlConnection("data source = 216.153.93.6,1092;" _
                        & "Initial catalog = db_hwy34Customs; uid=DaveT; pwd=Cobra")

        Dim cmd As New sc.SqlCommand
        Dim da As New sc.SqlDataAdapter
        Dim ds As New DataSet

        cmd.CommandType = CommandType.StoredProcedure
        cmd.CommandText = "dbo.spcc_AddCarsForSale"
        cmd.Connection = cnSQL

        cnSQL.Open()

        'cmd.Parameters.Add(Me.tbEmpName.Text, SqlDbType.Char, 50, "@tbEmpName")
        'cmd.Parameters.Add(Me.tbEmpPosition.Text, SqlDbType.Char, 50,
"@tbEmpPosition")
        'cmd.Parameters.Add(Me.tbEmpExperience.Text, SqlDbType.Char, 4,
"@tbEmpExperience")
        'cmd.Parameters.Add(Me.tbEmpSpecialization.Text, SqlDbType.Char, 50,
"@tbEmpSpecialization")
        'cmd.Parameters.Add(Me.tbEmpInfo.Text, SqlDbType.Char, 500, "@tbEmpInfo")
        'cmd.Parameters.Add(Me.tbEmpPicture.Text, SqlDbType.Char, 50,
"@tbEmpPicture")

        cmd.Parameters.Add("@tbCFSPicture", tbCFSPicture.Text)
        cmd.Parameters.Add("@tbCFSName", tbCFSName.Text)
        cmd.Parameters.Add("@tbCFSYear", tbCFSYear.Text)
        cmd.Parameters.Add("@tbCFSMake", tbCFSMake.Text)
        cmd.Parameters.Add("@tbCFSModel", tbCFSModel.Text)
        cmd.Parameters.Add("@tbCFSPrice", tbCFSPrice.Text)
        cmd.Parameters.Add("@tbCFSDescription", tbCFSDescription.Text)

        Try
            'cmd.ExecuteScalar() 'this is for updates or inserts
            cmd.ExecuteNonQuery()
```

```
        ' da.SelectCommand = cmd 'theses are for select statements()
        ' da.Fill(ds)
        ' ds.Tables(0).TableName = "CarsForSale"
        'ds.Tables(1).TableName = "SP_Email_List"

      Catch x As Exception

        lblComplete.Text = x.Message.ToString
        Exit Sub
      End Try

      'Go send the emails
      'ds = SendMail(ds)
      Response.Redirect("ControlCarsForSale.aspx")

      'lblComplete.Text = "Insert Complete"
    End Sub

End Class
```

ControlCarsForSale.aspx

```
Imports sc = System.Data.SqlClient

Public Class ControlCarsForSale
  Inherits System.Web.UI.Page

#Region " Web Form Designer Generated Code "

  'This call is required by the Web Form Designer.
  <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

  End Sub
  Protected WithEvents HyperLink1 As System.Web.UI.WebControls.HyperLink
  Protected WithEvents Label4 As System.Web.UI.WebControls.Label
  Protected WithEvents lnkAdminControls As System.Web.UI.WebControls.HyperLink
  Protected WithEvents Label5 As System.Web.UI.WebControls.Label
  Protected WithEvents lnkHome As System.Web.UI.WebControls.HyperLink
  Protected WithEvents Form2 As System.Web.UI.HtmlControls.HtmlForm
  Protected WithEvents lblErrorMessage As System.Web.UI.WebControls.Label

  'NOTE: The following placeholder declaration is required by the Web Form Designer.
  'Do not delete or move it.
  Private designerPlaceholderDeclaration As System.Object
```

```
  Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
    'CODEGEN: This method call is required by the Web Form Designer
    'Do not modify it using the code editor.
    InitializeComponent()
  End Sub

#End Region

  Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim cnSQL As New sc.SqlConnection("data source = 216.153.93.6,1092;" _
                    & "Initial catalog = db_hwy34Customs; uid=DaveT; pwd=Cobra")

    Dim cmd As New sc.SqlCommand
    Dim da As New sc.SqlDataAdapter
    Dim ds As New DataSet
    'Dim dtrDB As New sc.SqlDataReader

    cmd.CommandType = CommandType.StoredProcedure
    cmd.CommandText = "dbo.spcc_DisplayCarsForSale"
    cmd.Connection = cnSQL

    cnSQL.Open()

    Try
      'cmd.ExecuteScalar() 'this is for updates or inserts
      'cmd.ExecuteNonQuery()


      da.SelectCommand = cmd 'theses are for select statements()
      da.Fill(ds)
      ds.Tables(0).TableName = "CarsForSale"

    Catch x As Exception

      lblErrorMessage.Text = x.Message.ToString
      Exit Sub

    End Try

    Dim dtrDB = cmd.ExecuteReader()
    Response.Write("<table border=1>")
    Response.Write("<br>" & "<br>")
    Response.Write("<br>" & "<br>")
    Response.Write("<br>" & "<br>")
```

```
        Response.Write("<br>" & "<br>")
        'Response.Write("<br>" & "<br>")
        'Response.Write("<br>" & "<br>")

        While dtrDB.Read()
            Response.Write("<tr><td>" & " ")
            Response.Write("<a href =  /ImgDir/" & dtrDB(1) & "><img src=/ImgDir/" &
dtrDB(1) & " width=110 height=60></a>")
            Response.Write("<td>" & "Name: " & dtrDB(2) & "<br>" & "Year: " & dtrDB(3)
& "<br>" & "Make: " & dtrDB(4) & "<br>" & "Model: " & dtrDB(5) & "<br>" & "Price:
" & dtrDB(6) & "<br>" & "Description: " & dtrDB(7) & "</td>")
            Response.Write("<td>" & "<a href=editcarsforsale.aspx?FieldID=" & dtrDB(0) &
">Edit</a>" & "<br>" & "<br>" & "<a href=DeleteCarsForSale.aspx?FieldID=" &
dtrDB(0) & ">Delete</a>" & "</td>")
            'Response.Write("<td>" & dtrDB(4) & "</td>")
            'Response.Write("<td>" & dtrDB(5) & "</td>")
            'Response.Write("<td>" & dtrDB(6) & "</td>")
            'Response.Write("<td>" & dtrDB(7) & "</td></tr>")
        End While
        Response.Write("</table>")
        dtrDB.Close()
        cnSQL.Close()
    End Sub

    End Class
```

AddCarsForSale.aspx

```
Imports sc = System.Data.SqlClient
Public Class AddCarsForSale

    Inherits System.Web.UI.Page

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

    End Sub
    Protected WithEvents tbCFSDescription As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label5 As System.Web.UI.WebControls.Label
    Protected WithEvents tbCFSModel As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label4 As System.Web.UI.WebControls.Label
    Protected WithEvents tbCFSMake As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label3 As System.Web.UI.WebControls.Label
    Protected WithEvents tbCFSYear As System.Web.UI.WebControls.TextBox
```

```
    Protected WithEvents Label2 As System.Web.UI.WebControls.Label
    Protected WithEvents Label6 As System.Web.UI.WebControls.Label
    Protected WithEvents HyperLink2 As System.Web.UI.WebControls.HyperLink
    Protected WithEvents Label7 As System.Web.UI.WebControls.Label
    Protected WithEvents HyperLink3 As System.Web.UI.WebControls.HyperLink
    Protected WithEvents tbCFSPrice As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
    Protected WithEvents Label8 As System.Web.UI.WebControls.Label
    Protected WithEvents btnAddCFS As System.Web.UI.WebControls.Button
    Protected WithEvents Label9 As System.Web.UI.WebControls.Label
    Protected WithEvents tbCFSName As System.Web.UI.WebControls.TextBox
    Protected WithEvents lblComplete As System.Web.UI.WebControls.Label
    Protected WithEvents Button1 As System.Web.UI.WebControls.Button
    Protected WithEvents Label10 As System.Web.UI.WebControls.Label
    Protected WithEvents inpFileUp As System.Web.UI.HtmlControls.HtmlInputFile
    Protected WithEvents tbCFSPicture As System.Web.UI.WebControls.TextBox

    'NOTE: The following placeholder declaration is required by the Web Form Designer.
    'Do not delete or move it.
    Private designerPlaceholderDeclaration As System.Object

   Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Put user code to initialize the page here
    End Sub

    Private Sub btnAddCFS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddCFS.Click
        Dim cnSQL As New sc.SqlConnection("data source = 216.153.93.6,1092;" _
                        & "Initial catalog = db_hwy34Customs; uid=DaveT; pwd=Cobra")

        Dim cmd As New sc.SqlCommand
        Dim da As New sc.SqlDataAdapter
        Dim ds As New DataSet

        cmd.CommandType = CommandType.StoredProcedure
        cmd.CommandText = "dbo.spcc_AddCarsForSale"
```

```
        cmd.Connection = cnSQL

        cnSQL.Open()

        'cmd.Parameters.Add(Me.tbEmpName.Text, SqlDbType.Char, 50, "@tbEmpName")
        'cmd.Parameters.Add(Me.tbEmpPosition.Text, SqlDbType.Char, 50,
"@tbEmpPosition")
        'cmd.Parameters.Add(Me.tbEmpExperience.Text, SqlDbType.Char, 4,
"@tbEmpExperience")
        'cmd.Parameters.Add(Me.tbEmpSpecialization.Text, SqlDbType.Char, 50,
"@tbEmpSpecialization")
        'cmd.Parameters.Add(Me.tbEmpInfo.Text, SqlDbType.Char, 500, "@tbEmpInfo")
        'cmd.Parameters.Add(Me.tbEmpPicture.Text, SqlDbType.Char, 50,
"@tbEmpPicture")

        cmd.Parameters.Add("@tbCFSPicture", tbCFSPicture.Text)
        cmd.Parameters.Add("@tbCFSName", tbCFSName.Text)
        cmd.Parameters.Add("@tbCFSYear", tbCFSYear.Text)
        cmd.Parameters.Add("@tbCFSMake", tbCFSMake.Text)
        cmd.Parameters.Add("@tbCFSModel", tbCFSModel.Text)
        cmd.Parameters.Add("@tbCFSPrice", tbCFSPrice.Text)
        cmd.Parameters.Add("@tbCFSDescription", tbCFSDescription.Text)

        Try
            'cmd.ExecuteScalar() 'this is for updates or inserts
            cmd.ExecuteNonQuery()

            ' da.SelectCommand = cmd 'theses are for select statements()
            ' da.Fill(ds)
            ' ds.Tables(0).TableName = "CarsForSale"
            'ds.Tables(1).TableName = "SP_Email_List"

        Catch x As Exception

            lblComplete.Text = x.Message.ToString
            Exit Sub
        End Try

        'Go send the emails
        'ds = SendMail(ds)
        Response.Redirect("ControlCarsForSale.aspx")

        'lblComplete.Text = "Insert Complete"
    End Sub

End Class
```

EditCarsForSale.aspx

```
Imports sc = System.Data.SqlClient

Public Class CarsForSaleControl
   Inherits System.Web.UI.Page

#Region " Web Form Designer Generated Code "

   'This call is required by the Web Form Designer.
   <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

   End Sub
   Protected WithEvents tbCFSDescription As System.Web.UI.WebControls.TextBox
   Protected WithEvents Label5 As System.Web.UI.WebControls.Label
   Protected WithEvents tbCFSModel As System.Web.UI.WebControls.TextBox
   Protected WithEvents Label4 As System.Web.UI.WebControls.Label
   Protected WithEvents tbCFSMake As System.Web.UI.WebControls.TextBox
   Protected WithEvents Label3 As System.Web.UI.WebControls.Label
   Protected WithEvents tbCFSYear As System.Web.UI.WebControls.TextBox
   Protected WithEvents Label2 As System.Web.UI.WebControls.Label
   Protected WithEvents Label6 As System.Web.UI.WebControls.Label
   Protected WithEvents HyperLink2 As System.Web.UI.WebControls.HyperLink
   Protected WithEvents Label7 As System.Web.UI.WebControls.Label
   Protected WithEvents tbCFSPrice As System.Web.UI.WebControls.TextBox
   Protected WithEvents Label1 As System.Web.UI.WebControls.Label
   Protected WithEvents Label8 As System.Web.UI.WebControls.Label
   Protected WithEvents tbFieldID As System.Web.UI.WebControls.TextBox
   Protected WithEvents btnEditCFS As System.Web.UI.WebControls.Button
   Protected WithEvents tbCFSPicture As System.Web.UI.WebControls.TextBox
   Protected WithEvents lblComplete As System.Web.UI.WebControls.Label
   Protected WithEvents tbCFSName As System.Web.UI.WebControls.TextBox
   Protected WithEvents lblName As System.Web.UI.WebControls.Label
   Protected WithEvents HyperLink3 As System.Web.UI.WebControls.HyperLink
   Protected WithEvents Button1 As System.Web.UI.WebControls.Button
   Protected WithEvents inpFileUp As System.Web.UI.HtmlControls.HtmlInputFile
   Protected WithEvents Label9 As System.Web.UI.WebControls.Label

   'NOTE: The following placeholder declaration is required by the Web Form Designer.
   'Do not delete or move it.
   Private designerPlaceholderDeclaration As System.Object

   Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
      'CODEGEN: This method call is required by the Web Form Designer
      'Do not modify it using the code editor.
```

```
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        If Not Me.IsPostBack Then


            'Response.Write("<br>")
            'Response.Write(Request("FieldID"))

        Dim cnSQL As New sc.SqlConnection("data source = 216.153.93.6,1092;" _
                                    & "Initial catalog =db_hwy34Customs; uid=DaveT;
pwd=Cobra")


        Dim cmd As New sc.SqlCommand
        Dim da As New sc.SqlDataAdapter
        Dim ds As New DataSet
        'Dim dtrDB As New sc.SqlDataReader

        cmd.CommandType = CommandType.StoredProcedure
        cmd.CommandText = "dbo.spcc_DisplayCarsForSaleInTextBox"
        cmd.Connection = cnSQL


        cmd.Parameters.Add("@tbFieldID", Request("FieldID"))
        'tbFieldID.Text)
        '
        cnSQL.Open()

        Try
            da.SelectCommand = cmd 'theses are for select statements()
            da.Fill(ds)
            ds.Tables(0).TableName = "CarsForSaleToDisplay"


        Catch x As Exception


            lblComplete.Text = x.Message.ToString
            Exit Sub
```

```
        End Try


        Dim dtrDB = cmd.ExecuteReader()


        While dtrDB.Read()

            tbFieldID.Text = dtrDB(0)
            tbCFSPicture.Text = dtrDB(1)
            tbCFSName.Text = dtrDB(2)
            tbCFSYear.Text = dtrDB(3)
            tbCFSMake.Text = dtrDB(4)
            tbCFSModel.Text = dtrDB(5)
            tbCFSPrice.Text = dtrDB(6)
            tbCFSDescription.Text = dtrDB(7)

        End While
        dtrDB.Close()
            cnSQL.Close()

        End If

    End Sub

    Private Sub btnEditCFS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEditCFS.Click
        Dim cnSQL As New sc.SqlConnection("data source = 216.153.93.6,1092;" _
                            & "Initial catalog = db_hwy34Customs; uid=DaveT; pwd=Cobra")


        Dim cmd As New sc.SqlCommand
        Dim da As New sc.SqlDataAdapter
        Dim ds As New DataSet


        cmd.CommandType = CommandType.StoredProcedure
        cmd.CommandText = "dbo.spcc_EditCarsForSale"
        cmd.Connection = cnSQL


        cnSQL.Open()

        Response.Write("<br>" & "<br>")
        Response.Write("<br>")
        Response.Write("<br>")
```

```
        Response.Write("<br>")
        cmd.Parameters.Add("@tbFieldID", tbFieldID.Text)
        cmd.Parameters.Add("@tbCFSPicture", tbCFSPicture.Text)
        cmd.Parameters.Add("@tbCFSName", tbCFSName.Text)
        cmd.Parameters.Add("@tbCFSYear", tbCFSYear.Text)
        cmd.Parameters.Add("@tbCFSMake", tbCFSMake.Text)
        cmd.Parameters.Add("@tbCFSModel", tbCFSModel.Text)
        cmd.Parameters.Add("@tbCFSPrice", tbCFSPrice.Text)
        cmd.Parameters.Add("@tbCFSDescription", tbCFSDescription.Text)

        Try
            'cmd.ExecuteScalar() 'this is for updates or inserts
            cmd.ExecuteNonQuery()

            ' da.SelectCommand = cmd 'theses are for select statements()
            ' da.Fill(ds)
            ' ds.Tables(0).TableName = "MeetTheStaff"
            'ds.Tables(1).TableName = "SP_Email_List"


        Catch x As Exception

            lblComplete.Text = x.Message.ToString
            Exit Sub
        End Try

        'Go send the emails
        'ds = SendMail(ds)

        Response.Redirect("ControlCarsForSale.aspx")
        'lblComplete.Text = "Insert Complete"
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

    End Sub
End Class
```

DeleteCarsForSale.aspx

```
Imports sc = System.Data.SqlClient

Public Class DeleteCarsForSale
    Inherits System.Web.UI.Page
```

```
#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

    End Sub
    Protected WithEvents lblErrorMessage As System.Web.UI.WebControls.Label

    'NOTE: The following placeholder declaration is required by the Web Form Designer.
    'Do not delete or move it.
    Private designerPlaceholderDeclaration As System.Object

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim cnSQL As New sc.SqlConnection("data source = 216.153.93.6,1092;" _
                        & "Initial catalog = db_hwy34Customs; uid=DaveT; pwd=Cobra")

        Dim cmd As New sc.SqlCommand
        Dim da As New sc.SqlDataAdapter
        Dim ds As New DataSet

        cmd.CommandType = CommandType.StoredProcedure
        cmd.CommandText = "dbo.spcc_DeleteCarsForSale"
        cmd.Connection = cnSQL

        cnSQL.Open()
        cmd.Parameters.Add("@tbDelete", Request("FieldID"))
        'cmd.Parameters.Add("@tbDelete", tbDelete.Text)

        Try

            cmd.ExecuteNonQuery()

        Catch x As Exception

            lblErrorMessage.Text = x.Message.ToString
            Exit Sub
```

```
        End Try

        Response.Redirect("ControlCarsForSale.aspx")
    End Sub

End Class
```

Project Plan Approval Form

Bill DeKnikker
Final Project Planning
INFS 788-1

Introduction:
My project is to build a web site with a database that can be edited using a web-based interface. The client is an auto body shop that specializes in custom hot rods (a confirmed interest of mine).

Unlike typical auto body shops that merely fix customer's automobiles, this body shop does custom work on a variety of hot rods and classic car restoration. They also build their own hot rods and restorations, which they sell to the public and collectors around the country.

Rather than a static web site, this web site will have a database that anyone with administration rights can access and edit through a web-based interface. This client has a product line that changes periodically and these changes need to be updated through a user interface.

Statement of problem or question you have identified and brief summary of current situation:
This client wants to have the ability to add inventory, edit inventory entries, delete inventory as necessary and add/delete pictures associated with the inventory. These data anomalies need to occur via a web-based user interface by someone without working knowledge of databases, HTML skills or direct FTP access to the server files.

My client would like me to build a web site they can maintain independently by editing their own inventory and uploading their own pictures.

The database will store information about the inventory such as the year, make and model of vehicles currently owned by the company, works in progress and past projects. Text area will be provided for the client to write descriptions of the projects and narrations of before & after photos.

Goals, objectives, purpose:
My goals, objectives and purpose include, designing, building and publishing a web site for Hwy 34 Customs body shop. I will mostly use Macromedia's DreamWeaver studio to create the HTML web page. I will use DreamWeaver as an HTML editor and Fireworks (and possibly Flash) for some of the graphics. I will also use Adobe Photoshop for some of the graphics.

For the database, I have chosen MySQL because of its availability and the scope of this project. Security is not much of a concern with this project and the database will be small.

I will use ASP.net to connect the web site to the database and provide the functionality. I am considering adding at least one form to the web site and using PHP to process the form. However, no need for a form has been determined.

Scope of Work, Plan of Action, Activities:
The owner of the web site will have the ability to upload pictures and edit the product line via a web-based interface. Client interaction would include the ability to add/delete pictures and descriptions for such products as:

- Work in progress
- Custom restorations
- Cars for sale
- Custom hot rods for sale

Work Breakdown Structure:
The first thing on the agenda, which is beyond the scope of this project, is to install and configure a server, which will host the web site. An IP address will need to be obtained as well as a domain name. (There will actually be two domain names hosted on this server.)

The next step is to actually build and publish the web site. The initial web site will be static until the database and web interface are set up. The reason for this sequence of events is to establish a web presence for the client as quickly as possible.

Next, the database will be designed, built and uploaded onto the web server. At that point, a user access point will be setup with password protection. Then the web interface will be created and users with administrative rights will be able to access the web interface for data management.

Time line:

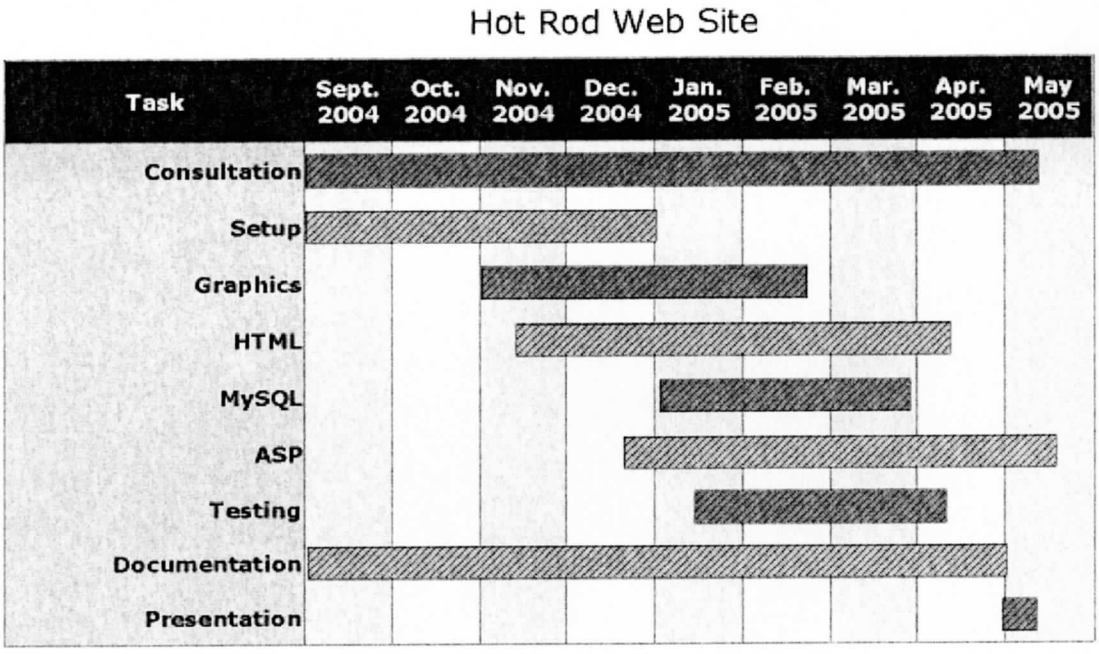
I am currently in the process of setting up and configuring a server to host this web site (and another one owned by the same client). As web presence is desired as soon as possible, I plan to establish web presence by the beginning of the Spring 2005 semester, which is when my actual project implementation course officially begins. I need to have the entire project completed by May 2005.

September-October 2004: Consultations with the client for the development of the web site. Primary concerns include the actual design of the web site aesthetically as well as functionally. Submit project proposal plan approval form for course completion.

November-December 2004: Continue setting up IIS on Windows Server 2003 for web hosting. Complete graphic design and HTML coding to establish web presence.

January-March 2005: Install and configure MySQL for database. Start coding HTML administration interface for editing capabilities. Begin .ASP coding for database update anomalies.

April-May 2005: Continue .ASP coding, test code and database connection. Verify web site integrity and functionality. Populate database. Prepare presentation for MSIS graduation. Present.



Deliverables:
The deliverables for this project will consist of a client web site with a database and web interface for editing the database.